

XV. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2019. január 24–25.

Különböző függőségi elemzők teljesítményének vizsgálata magyar nyelven

Tálas Dalma¹, Novák Attila^{1,2}¹Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Kar²MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport

Budapest, Práter u. 50/a.

talasdalmaalexandra@hotmail.com, novak.attila@itk.ppke.hu

Kivonat Cikkünkben összehasonlítjuk néhány különböző elven működő függőségi elemző magyar szövegkorpuszon nyújtott teljesítményét. Emellett bemutatjuk, hogy a szövegkorpusz annotációjának akár teljesen automatizált javításával teljesítménybeli javulás érhető el az annotációban használt címkekészlet felbontásának növelése mellett is.

1. Bevezetés

A függőségi elemzés a mondatelemzés egy fajtája, amely során azt vizsgáljuk, hogy a mondatban lévő szavak milyen kapcsolatban állnak egymással. Ezeket a kapcsolatokat irányított élekkel írjuk le úgy, hogy egy szóba csak egy él mutathat, de kifelé bármennyi él mehet. A függőségi elemzésre különböző gépi tanulási algoritmusok léteznek, amelyek alapulhatnak valamilyen neurális hálózaton, vagy használhatnak egyéb lineáris vagy nemlineáris módszereket.

Az elemzéshez szükség van nagyméretű, annotált szövegkorpuszra. A legnagyobb, manuálisan ellenőrzött függőségi elemzést is tartalmazó magyar szövegkorpusz a Szeged Dependency Treebank [1]. Probléma azonban, hogy az ebben alkalmazott annotációs sémában számos egymástól meglehetősen különböző szintaktikai szerkezet annotációja nem különbözik a használt függőségi relációk szintjén (pl. jelzők, birtokosok és mellékmondatok), vagy valamilyen egyéb az annotáció megtervezésénél tett megfontolás nehezíti a szerkezetek értelmezését, illetve olyan manuálisan beszűrt elemeket tartalmaz (például a névszói állítmányok mellett feltételezett zérus létigéket), amelyek az eredeti korpuszban nem szerepelnek, és nem is áll rendelkezésre megfelelő gépi modell az ilyen elemek beszűrésére a függőségi elemzés folyamán.

A Universal Dependencies (UD) projekt¹ célja, hogy kiküszöbölje vagy lehetőleg minimalizálja a különböző nyelvek függőségi elemzésére használt annotációs sémák közötti idioszinkratikus eltérésekből adódó azon hatást, hogy nem vagy nagyon nehezen összehasonlíthatóak a különböző – akár nagyon közeli rokoni kapcsolatban álló – nyelvekhez készült függőségi elemzést tartalmazó treebankek,

¹ <http://universaldependencies.org>

és így a különböző nyelvekben előforduló szintaktikai szerkezetek is. Célja, hogy egy olyan függőségi annotációs szabályrendszert alkosson meg, amely minél nagyobb mértékű nyelvészeti konszenzuson alapul, minél könnyebben értelmezhető az emberek számára, és minél helytállóbb módon és egységesen írja le a különböző nyelvek sokszor nagyon eltérő szerkezeteit is. Az ideális tanítókörpuszban egyesülne a minőség és a mennyiség, azaz a Szeged Dependency Treebank mérete és a Universal Dependencies korpusz logikus és átlátható elemzési módszere. Az itt bemutatott munkában nem valósítottuk meg a Szeged Dependency Treebank UD 2.0 formátumra hozását. Célunk pusztán egy olyan annotáció automatikus létrehozása volt, amely – miközben összehasonlítható marad az eredetivel – annál pontosabban azonosít bizonyos szerkezeteket, a gép számára mégis jól tanulható. Ehhez az UD specifikációjából merítettünk ihletet. Ugyan az UD-nek része a Szeged Treebank egy konvertált kis részlete, az ebben alkalmazott annotáció sem felel meg pontosan a kurrens UD 2.0 specifikációnak.

A függőségi elemzés kiértékelését három metrika alapján végeztük. A *Label Accuracy* (LA) csak a címke, az *Unlabeled Attachment Score* (UAS) csak az él, és a *Labeled Attachment Score* (LAS) az él és a címke együttes egyezését vizsgálja.

2. Kapcsolódó munkák

Három különböző elemző teljesítményét vizsgáltuk meg. A MateParser [2] egy gráf-alapú, support vector machine módszert használó elemző, amelyet 2010-ben fejlesztettek ki. A SyntaxNet [3] egy átmenet-alapú, neurális hálózaton alapuló elemző, amelyet a Google fejlesztett ki és tett publikussá 2016-ban. Az Parser v2.0 (eredeti nevén Unstable Parser) [4] egy neurális hálózaton alapuló, gráf-alapú algoritmus, amelyet a Stanford Egyetemen fejlesztettek ki, és amely megnyerte a 2017-es CONLL függőségi elemzési versenyfeladatot az összes, feladatkiírásban szereplő nyelvre, és 2018-ban is egy ugyanezen az elemzőn alapuló rendszer lett a nyertes. Az utóbbi két algoritmus a függőségi elemzőn kívül magában foglal egy szófaji egyértelműsítő és egy morfoszintaktikai elemzőt is.

A kézi függőségi annotációt tartalmazó Szeged Dependency Treebank [1] 82 ezer mondatból áll, ami 1,2 millió szónak felel meg. Megtalálhatók benne különböző újságcikkek, informatikai és jogi szövegek, 14-16 éves tanulók írásai, üzleti és pénzügyi szövegek és fiktív történetek; tehát sokféle doménből származó szövegek. A treebankben használt függőségi címkék teljes listája megtalálható az [1]-es cikkben.

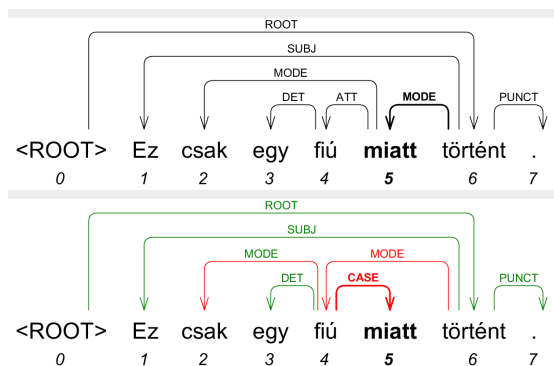
A modellek által készített elemzések kiértékeléséhez, az átalakítások ellenőrzéséhez, és az összehasonlító ábrák készítéséhez a MaltEval nevű programot használtuk [5]. A MaltEval alkalmas mind számszerű kiértékelésre (fedés, pontosság, F-mérték mutatójára), mind vizuális megjelenítésre.

3. Az annotáció minőségének javítása

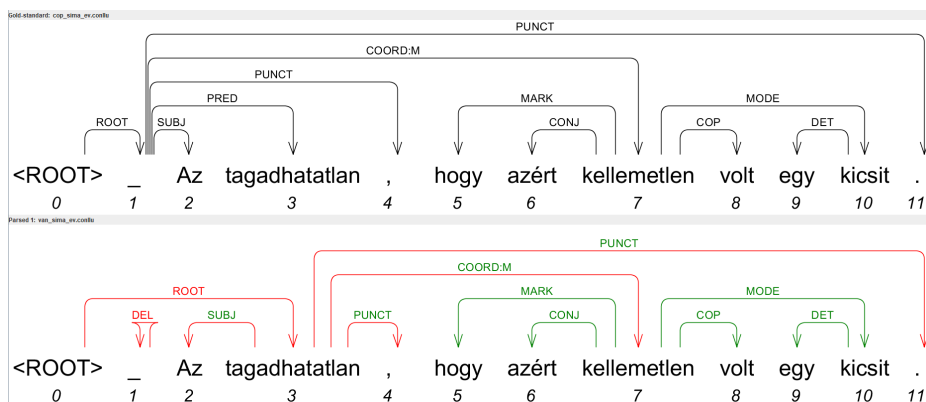
A Szeged Dependency Treebank annotációjában előfordulnak különböző típusú hibák. Egyrészt szerepelnek véletlen hibák, elgépelések, másrészt előfordul, hogy

következetlen egy-egy elemzés, amely nem követi a korpuszban egyébként alkalmazott annotációs sémát, harmadjára pedig vannak szerkezetek, ahol az adott szerkezet elemzési stratégiája nem tűnik a legintuitívabb megoldásnak. Ezen hibák automatikus módszerekkel javítható eseteit próbáltuk meg megtalálni és kijavítani. Emellett az automatikusan szűrhető, de nem javítható hibák esetében megvizsgáltuk, hogy milyen teljesítményt kapunk egy automatikus minőségi szűrés alkalmazásával létrehozott részkorpuszon.

Az annotáció átírása során a Universal Dependencies projektben használt annotációs elvekből merítettünk ihletet, bár az egyszerűség és az eredetivel való könnyebb összehasonlíthatóság érdekében nem tértünk át az ott alkalmazott függőségi címkékre. Hogy az *az a kutya* típusú predeterminánsos szerkezetekben a mutató névmási determinánst is az NP fejéhez kapcsolhassuk, de mégse kapjon azonos DET címkét a két determináns, a mutató névmás számára külön címkét definiáltunk ezekben a szerkezetekben. A névutós szerkezetek fejének a névszót tettük meg (a névutó helyett) és külön címkét definiáltunk a névutóra (1. ábra). A többszörös ROOT címkéket megszüntettük. Az ATT címkét, amit korábban sok egymástól eléggé különböző viszony jelölésére használtak, alcímkékre bontottuk. Bevezettünk minőség-, mennyiség- és birtokos jelzői címkéket, megszüntettük az ATT címke névutós szerkezetekben való használatát, és a tagmondatokat összekötő ATT címkét helyettesítettük a tagmondatok közötti viszonyt leíró címkékkel (2. ábra). A fráziskoordinációk elemzését átalakítottuk úgy, hogy a frázisok egymással legyenek összekötve, és ne a kötőszón keresztül, továbbá a szerkezet fejének az utolsó frázist tettük meg az első helyett (ebben eltértünk az UD specifikációtól is). A névszói-igei állítmányok esetében a szerkezet fejének a névszót választottuk, amihez kapcsolódik az ige az újonnan bevezetett COP címkén keresztül. Az annotációban korábban használt, a testetlen létige jelölését szolgáló tokeneket kivettük és a megmaradó szavak viszonyait megfelelően újrageneráltuk (2. ábra). Ez utóbbi javítás volt talán a legfontosabb ahhoz, hogy a kapott modell nyers szövegre alkalmazva is működőképes maradjon.



1. ábra. Példa a névutós szerkezetek annotációjának átalakítására



2. ábra. Példa a testetlen létige és az alárendelő mondat annotációjának átalakítására (a törlendő token az illusztráció érdekében szerepel DEL címkével)

4. A tanítás és a tesztelés folyamata

A neurális hálózaton alapuló elemzők tanításához (SyntaxNet és Parser v2.0) három szöveghalmazra volt szükségünk: tanító-, validációs és teszt-halmazra, ezért ezekhez a korpuszt 8:1:1 arányban osztottuk fel. A MateParser esetében nem volt szükség validációs halmazra, ezért itt kétfelé osztottuk a korpuszt 9:1 arányban úgy, hogy a teszt-halmaz teljes mértékben egyezzen a többi elemző tesztelésére használt halmazzal, a tanítókorpusz pedig magában foglalja a validációs halmazt is. A korpusz felosztásánál figyeltünk rá, hogy mindegyik halmaz reprezentatív legyen, azaz pl. a teszt-halmaz ne tartalmazzon doménen kívüli szövegeket.

A függőségi elemzők jelenleg két menetben végzik a beadott nyers szöveg elemzését. Első lépésként morfoszintaktikai annotációt (és esetleg lemmatizálást) végeznek a nyers szövegen, majd ennek eredményét használják a függőségi annotációt végző parser bemeneteként. Jelen kutatásunkban nem vizsgáltuk az elemzők morfoszintaktikai annotációt végző címkéző (tagger) komponensének teljesítményét, kizárólag magára a szintaktikai elemzőre koncentráltunk. Mindegyik elemző a gold standard morfoszintaktikai annotációt használta bemeneteként.

5. Eredmények

5.1. A függőségi elemzők eredményeinek összevetése

Először a módosítás nélküli eredményeket vizsgáltuk meg, azaz a Szeged Treebank eredeti annotációjával tanítottuk az elemzőket. A teljesítményt megvizsgáltuk a morfoszintaktikai jegyek nélkül is, tehát úgy, hogy csak a szót, a szótövet és a szófajcímkét adtuk oda az elemzőnek a függőségi paramétereken kívül, és úgy is, hogy ezek mellé a morfoszintaktikai jegyeket is hozzáfűztük (1. táblázat).

	MateParser			SyntaxNet			Parser v2.0		
	LA	UAS	LAS	LA	UAS	LAS	LA	UAS	LAS
Jegyek nélkül	0,931	0,921	0,882	0,848	0,866	0,768	0,969	0,895	0,877
Morfosz. jegyekkel	0,955	0,932	0,908	0,907	0,916	0,845	0,973	0,900	0,884

1. táblázat. A három függőségi elemző teljesítménye az eredeti annotáción három metrika alapján

Az 1. táblázatban láthatók a három függőségi elemző által elért pontosságok három különböző metrika alapján. Azok a modellek, amelyek a morfoszintaktikai jegyeket nem használták a tanításhoz, minden esetben rosszabb pontosságot értek el, mint a morfoszintaktikai jegyeket használó modellek. Az eredményekből az is megállapítható, hogy összességében véve – azaz LAS metrika alapján – a jegyek nélküli és a jegyeket használó modellek esetén is a MateParser érte el a legjobb teljesítményt. Az él helyét illetően is a MateParser bizonyult a legpontosabb elemzőnek, azonban érdekes, hogy a címkék szerinti metrika alapján a Parser v2.0 érte el a legjobb eredményeket mind jegyek nélkül, mind azokkal. A SyntaxNet teljesítménye elmaradt a másik két elemzőétől. Ennek fő oka az lehet, hogy a SyntaxNet nem gráfolapú elemző, és nem tud mit kezdeni az olyan nem projektív szerkezetekkel, ahol a függőségi élek keresztezik egymást.

	Régi annotáció			Új annotáció			Relatív javulás (%)		
	LA	UAS	LAS	LA	UAS	LAS	LA	UAS	LAS
MateParser	0,955	0,932	0,908	0,964	0,934	0,919	20,00	2,941	11,96
Parser v2.0	0,973	0,900	0,884	0,968	0,942	0,927	-18,52	42,00	37,07

2. táblázat. A régi és az új annotáción tanított modellek pontossága és a relatív javulás mértéke három metrika szerint

5.2. Az átalakított annotáción kapott eredmények

Az annotáció átalakítása után újabb modelleket tanítottunk be a teljes korpuszon. Az elemzők közül csak a MateParserrel és az Parser v2.0-val dolgoztunk a továbbiakban, mert a SyntaxNet teljesítménye elmaradt a többi elemzőétől.

A MateParserrel tanított modell eredményein látszik, hogy az új annotáció minden esetben javított a pontosságon (2. táblázat). A javulás mértéke a címkék esetén volt a legnagyobb: a címkehibák 20%-át és az élhibák 3%-át sikerült elkerülni az új annotáción tanított modellnek.

A Parser v2.0-val tanított modell esetében az élek szerinti pontosság nőtt nagy mértékben, míg a címkék szerinti pontosság csökkent (2. táblázat). Összességében véve (LAS metrika alapján) azonban a teljesítmény így is sokat javult.

A régi annotáción a Parser v2.0 a címkéket jobban jósolta, mint a MateParser, míg az éleket rosszabbul, és összességében véve kicsit pontatlanabbul jósolt. Az új annotáción azonban minden metrika szerint jobb eredményt ért el, mint a MateParser.

Érdekes, hogy a Parser v2.0-nál az élek jóslásán sikerült sokat javítani – ahogy arra számítani lehetett –, a MateParser-nél viszont pont a címkék szerinti eredmény lett jobb. Ezért érdemes megvizsgálni az elért pontosságokat címkék szerinti bontásban is.

A régi és az új annotáción tanított modellek összehasonlításához megvizsgáltuk a címkék szerinti pontosságokat is, ami itt különösen fontos volt, hiszen az átalakítások során nagymértékben módosítottunk a címkehalmazon (3. táblázat). Az érintetlenül hagyott címkék közül az új annotáció – közvetett módon – jelentősen javított a DAT, a DET és a NEG címkéken. A 3-3 hely- és időhatározót jelölő címkén hol sikerült javítani, hol nem, de a következtetlen annotáció miatt (a korpusz legnagyobb részében csak névmások kaptak ilyen annotációt, helyenként azonban a hely- vagy időhatározói névszói csoportok feje is) ezeket a címkéket nem érdemes figyelni az automatikus kiértékelés során. Néhány címke, mint pl. az INF és a ROOT, pontossága romlott. Ezek közül a mondat fejének megtalálása különösen fontos lenne. A ROOT jóslása azért romolhatott, mert a régi annotációban sok helyen a külön beszúrt VAN token jelentette a mondat fejét, amit persze sokkal egyszerűbben meg tudott találni az algoritmus, ugyanakkor valós helyzetekben ilyen hozzáadott annotáció nem áll rendelkezésre. Ezeken kívül voltak olyan címkék, amelyeknek pontossága vagy az egyik, vagy a másik elemzónél javult vagy éppen romlott.

Ami a szétbontott címkéket illeti, az eredmények változóak voltak. A négy eredeti címke (COORD, CONJ és ATT) és a PRED pontossága sokat romlott, ami várható volt, hiszen ezeknek teljesen meg kellett volna szűnniük (a CONJ kivételével), viszont az automatikus átalakítás nehézségei és az eredeti annotációban szereplő hibák miatt még maradtak ilyen címkék is a korpuszban. Ehhez képest a pontosság még így is viszonylag magas, amiből arra következtettünk, hogy az annotációban maradt címkék olyan szerkezetekben szerepelnek, amelyek vagy amelyeknek egy része valamilyen közös mintát mutat. Az alcímkéket nagyon jól sikerült megjósolnia a modellnek a birtokos, a minőség- és a mennyiségjelzős szerkezetekben. Ezzel szemben a főnévi ATT módosítók eredményei rosszabbak lettek, ami lehetett egyrészt azért, mert sok ilyen szerkezet valójában hibás különírást tartalmazott, aminek a szófajcímkéi és esetleg egyéb annotációi hiányosak voltak, másrészt lehetett azért, mert a modell összekeverte a tulajdonneves szerkezetekkel.² A négy alcímke közül három pontossága jobb, mint az eredeti címkéé volt a régi annotáción betanított modell kimenetében. Az átalakítás során az ATT címkék egy másik részéből CASE lett (a névutós szerkezetekben), amit nagyon jól sikerült jósolnia a modellnek, de ez várható is volt. A frázis- és a

² Az UD magyar részkorpusza éppen abban nem felel meg az UD 2.0 specifikációnak, hogy az utóbbi szerint a szabályos szintaktikai szerkezetet tartalmazó névelemeket (például a címekeket) a szokásos függőségi címkék használatával kellene annotálni – ez azonban már a Szeged Dependency Treebankben sincs így

	MateParser		Parser v2.0	
	Régi	Új	Régi	Új
APPEND	0,874	0,861	0,894	0,878
ATT	0,956	0,782	0,974	0,792
ATT:A		0,990		0,990
ATT:M		0,989		0,989
ATT:N		0,911		0,921
ATT:POSS		0,975		0,976
AUX	0,989	1,000	1,000	0,996
CASE		0,989		0,990
CC		0,956		0,957
CONJ	0,972	0,945	0,995	0,951
COORD	0,882	0,582	0,918	0,580
COORD:C		0,897		0,913
SUBORD		0,905		0,920
COORD:P		0,894		0,903
COP		0,906		0,928
DAT	0,889	0,935	0,933	0,936
DET	0,991	0,998	0,995	0,999
FROM	0,691	0,661	0,786	0,765
INF	0,989	0,978	0,993	0,982
IS		0,996		0,996
LOCY	0,827	0,819	0,860	0,836
MARK		0,979		0,978
MODE	0,895	0,916	0,925	0,919
NE	0,928	0,993	0,995	0,994
NEG	0,979	0,992	0,995	0,995
NUM	0,990	0,989	0,991	0,990
OBJ	0,973	0,981	0,987	0,984
OBL	0,961	0,973	0,975	0,975
PRED	0,862	0,509	0,908	0,458
PREDET		0,965		0,972
PREVERB	0,973	0,993	0,994	0,993
PUNCT	1,000	1,000	1,000	1,000
QUE	0,950	0,929	0,926	0,933
ROOT	0,967	0,949	0,982	0,962
SUBJ	0,921	0,947	0,962	0,958
TFROM	0,878	0,857	0,835	0,825
TLOCY	0,892	0,899	0,900	0,893
TO	0,751	0,790	0,825	0,827
TTO	0,773	0,815	0,787	0,796

3. táblázat. A régi és az új annotáción tanított modellek F-mértéke címkék szerinti lebontásban

mondatkoordináció felismerése is javult valamennyit, a kötőszavak megtalálása viszont romlott. Persze ezt nehéz összevetni a régebbi annotáció eredményeivel, mert megváltozott a szerkezetek elemzési logikája, viszont az látszik, hogy a tagmondatok közötti – azon belül is az alárendelő – kötőszavak jóslása könnyebben ment az elemzőnek, mint a fráziskoordináció esetében. A névszó-igei állítmány felismerése nem triviális feladat, a COP címke jóslási pontossága mégis viszonylag magas.

5.3. A szűrt részkorpuszon kapott eredmények

Az új annotáción kapott címkék szerinti pontosságokon látszik, hogy azok a címkék, amelyek a hibák miatt bennmaradtak a korpuszban, de amelyeket szeretnénk teljesen megszüntetni, nagyon sokat rontottak a pontosságokon. Ebből adódik az ötlet, hogy ki lehetne szűrni a halmazokból az ilyen módon hibásnak érzékelt mondatokat, és be lehetne tanítani egy olyan modellt, amely csak a megmaradt, látszólag jobb minőségű annotációval ellátott mondatokat használja. A kérdés az, hogy ha az egy-egy szónál előforduló rossz annotáció miatt kivesszük a teljes mondatot a tanítóhalmazból, akkor azzal inkább ártunk a modellnek, azaz lehet, hogy volt egy hiba az egyik szónál, de a mondat többi része még így is túl sok értékes információt hordozott; vagy inkább segítünk a modellnek, azaz a hibás mondatok annyira zavaróak a tanítás számára, hogy inkább zajt jelentenek, így a kihagyásukkal többet nyernénk, mint amennyi információ elvész.

A korpuszból kiszűrtük azokat a mondatokat, amelyek hiányos annotációval rendelkeztek a szófajcímkét illetően ($posTag = X, Y, Z$), vagy amelyekben az átalakítás után is szerepelt teljesen megszüntetendő függőségi címke (ATT, COORD vagy PRED). Az így kapott korpusz mérete (mondatszám alapján) az eredeti 83,7%-a lett. A szűrt korpuszon kapott eredmények jobbak, mint a teljes korpuszon tanított modell esetében (4. táblázat). A MateParserrel 15,7%-os, a Parser v2.0-val 13,7%-os relatív javulást értünk el. A kapott pontosságok magasabbak, mint az adott elemzővel tanított bármelyik másik modellé. Ezen belül az Parser v2.0 teljesítménye volt a legjobb, LAS metrika esetén 0,937.

	Teljes korpuszon tanítva		Szűrt korpuszon tanítva		Javulás mértéke	
	Mate	P2.0	Mate	P2.0	Mate	P2.0
LA	0,964	0,968	0,970	0,973	16,67%	18,62%
UAS	0,934	0,942	0,944	0,950	15,15%	13,80%
LAS	0,919	0,927	0,930	0,937	15,71%	13,70%

4. táblázat. A szűrt halmazon tanított modell pontossága és relatív javulása a teljes korpuszon tanított modellhez képest

6. Konklúzió

Vizsgálatunk során a MateParser és a Parser v2.0 kiemelkedően jó eredményeket ért el. Az eredeti korpuszon az előbbi az élek jóslása terén volt jobb, míg az utóbbi a címkék eltalálásában. A címkék szerinti pontosságokból pedig az látható, hogy teljesítmény másképpen oszlik el a két elemző esetében. A SyntaxNet eredményei minden vizsgált esetben elmaradtak a másik két elemzőétől.

A vizsgálat során azt is megállapítottuk, hogy az automatikusan javított új annotáción tanított modellek jobban teljesítenek, mint a korábbiak. Elemzőtől függően más-más helyeken tapasztaltunk javulást, pl. a MateParsernél elsősorban a címkéken sikerült javítani, míg a Parser v2.0-nál az éleken. Összességében véve az új annotáción tanított modellek a MateParser esetében 10%-os, a Parser v2.0 esetében 37%-os relatív javulást értek el a régi annotáción tanított modellekhez képest. Az újonnan bevezetett címkék jóslása általában jobban ment a modelleknek, mint a megfelelő régi címkéké annak ellenére, hogy a lehetőségek száma nőtt. Ez igazolja azt az várakozást, hogy intuitívebb és konzisztensebb annotáció alapján a gép is könnyebben tanul meg elemezni. A korábban egybemosott kategóriák szétválasztása nem ártott a rendszernek, inkább javult a pontosság.

Hasonló eredmény olvasható ki Simkó et al. cikkéből is [6], bár abban a kutatásban épp ellenkező irányban módosították a címkekészletet, mint mi az itt leírt kísérleteinkben: a magyar UD korpusz címkekészletének elemeit összevonva általában romlottak a címkézési pontosságok (LAS, UAS) amellett, hogy a címkék összevonása még információvesztéssel is járt.

A minőségi szempontból szűrt halmazon tanított modellek teljesítménye jobb, mint bármelyik másik halmazon vagy annotációval tanított modellé (itt már csak a pontosabban működő MateParser és Parser v2.0-t vizsgáltuk). A címkék szerinti pontosság 97 és 97,3%-os, az élek szerinti 94,4 és 95%-os, míg az élek és címkék együttes helyességét tekintve a teljesítmény 93, illetve 93,7%-os, mindhárom metrika szerint a Parser v2.0 javára. Tehát összességében véve ez a két modell érte el a legjobb teljesítményt, illetve azon belül a Parser v2.0-val tanított elemző volt a legpontosabb.

Köszönetnyilvánítás

Jelen kutatás az FK 17 pályázati program finanszírozásában az FK 125217 számú projekt keretében a Nemzeti Kutatási Fejlesztési és Innovációs Alapból biztosított támogatással valósult meg.

Hivatkozások

1. Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z., Csirik, J.: Hungarian dependency treebank. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (2010)

2. Björkelund, A., Bohnet, B., Hafdell, L., Nugues, P.: A high-performance syntactic and semantic dependency parser. In: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations. COLING '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 33–36
3. Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M.: Globally normalized transition-based neural networks. CoRR **abs/1603.06042** (2016)
4. Dozat, T., Qi, P., Manning, C.D.: Stanford’s graph-based neural dependency parser at the CoNLL 2017 shared task. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Vancouver, Canada, August 3-4, 2017. (2017) 20–30
5. Nilsson, J., Nivre, J.: Malteval: an evaluation and visualization tool for dependency parsing. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, B.M.J.M.J.O.S.P.D.T., ed.: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08), Marrakech, Morocco, European Language Resources Association (ELRA) (2008) <http://www.lrec-conf.org/proceedings/lrec2008/>.
6. Simkó, K.I., Kovács, V., Vincze, V.: Szintaktikai címkekészletek hatása az elemzés eredményességére. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 20117), Szeged, SZTE (2017) 316–322